Pages 1 - 96

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

BEFORE THE HONORABLE WILLIAM H. ALSUP

ORACLE AMERICA, INC., Plaintiff,	)	
VS.	)	No. C 10-3561 WHA
GOOGLE, INC., Defendant.	)	Wednesday San Francisco, California February 24, 2016

TRANSCRIPT OF PROCEEDINGS

APPEARANCES:

For Plaintiff:

ORRICK, HERRINGTON & SUTCLIFFE
51 West 52nd Street
New York, New York 10019
BY: PETER A. BICKS, ESQ.
LISA T. SIMPSON, ESQ.

ORRICK, HERRINGTON & SUTCLIFFE
405 Howard Street
San Francisco, California 94105
BY: ANNETTE L. HURST, ESQ.

ORRICK, HERRINGTON & SUTCLIFFE
1000 Marsh Road
Menlo Park, California 94025
BY: GABRIEL M. RAMSEY, ESQ.

(APPEARANCES CONTINUED ON FOLLOWING PAGE)

Reported By: ***Debra L. Pas, CSR 11916, CRR, RMR, RPR***
Official Reporter - US District Court
Computerized Transcription By Eclipse

APPEARANCES: (CONTINUED)

For Plaintiff:
ORACLE USA, INC.
Legal Department
500 Oracle Parkway
Redwood Shores, California 94065
BY: DEBORAH KAY MILLER, ESQ.
MATTHEW M. SARBORARIA, ESQ.
ANDREW TEMKIN, ESQ.
RUCHIKA AGRAWAL, ESQ.

For Defendant:
KEKER & VAN NEST
633 Battery Street
San Francisco, California 94111
BY: ROBERT ADDY VAN NEST, ESQ.
CHRISTA MARTINE ANDERSON, ESQ.
MICHAEL S. KWUN, ESQ.
EUGENE MORRIS PAIGE, ESQ.
REID PATRICK MULLEN, ESQ.
EDWARD ANDREW BAYLEY, ESQ.
MAYA BETH KARWANDE, ESQ.

GOOGLE, INC.
1600 Amphitheatre Parkway
Mountain View, California 94043
BY: CATHERINE LACAVERA, ESQ.
RENNY F. HWANG, ESQ.

For Dr. Kearl:
FARELLA, BRAUN & MARTEL LLP
235 Montgomery Street, 30th floor
San Francisco, California 94104
BY: JOHN L. COOPER, ESQ.

- - -

P R O C E E D I N G S

FEBRUARY 24, 2016 **7:56 A.M.**

THE CLERK: Calling Civil Action 10-3561, Oracle America, Inc. versus Google, Inc. Matter is on for a tutorial hearing this morning.

Counsel, can you please state your appearances for the record.

MR. VAN NEST: Good morning, your Honor. Bob Van Nest from Keker and Van Nest for Google.

I'm here with Christa Anderson, Michael Kwun, Gene Paige, Reid Mullen, Ed Bayley and Maya Karwande.

THE COURT: All right. Welcome.

MR. VAN NEST: We also have from -- from Google, we have Catherine Lacavera and Renny Hwang. Good morning.

THE COURT: Okay. Welcome to you all. Good morning.

MR. BICKS: Good morning, your Honor. Peter Bicks from Orrick for Oracle.

My partner, Annette Hurst.

MS. HURST: Good morning, your Honor.

MR. BICKS: Gabe Ramsey. Lisa Simpson.

MS. SIMPSON: Good morning, your Honor.

MR. BICKS: And we have our client back here; Deborah Miller, Andrew Temkin and Matt Sarboraria and Ruchika Agrawal.

THE COURT: Thank you all. Thank you for coming.

And?

MR. COOPER: John Cooper, your Honor, for Dr. Kearl.

THE COURT: Okay. Welcome to you, Mr. Cooper.

And, again, the world should know that Mr. Cooper has volunteered his services without charge. He's doing this for the good of the order and I can't say how grateful the Court is to Mr. Cooper and his firm for being such a good citizen.

So once again, Mr. Cooper, thank you.

MR. COOPER: Thank you, your Honor.

THE COURT: Okay. I apologize that we had to switch courtrooms. We discovered yesterday getting ready for today's hearing that the electronic system had a glitch and we could not figure out what it was, and Judge Henderson was good enough to let me use his courtroom for today. So we'll get it fixed in due course.

So I received -- we're going to go right to this tutorial part, but if time permits -- and I'm not saying that it will, but if time permits, at the end of today's hearing I would like to address the February 23rd letter that I got from Oracle about Google not letting their -- the other side's expert see certain documents and the expert not disclosing who he had worked for earlier. So possibly we'll get into that.

But the reason I'm bringing it up now is that I want you in the meantime, with the gigantic teams, I want the stack of documents in issue to be handed up to me in a couple of hours so that I can look at them and I may just re-designate them on

the spot.

So that is what you're up against when you're dealing with me, is I'm going to cut through the BS and I'm going to see if these documents still deserve to be counted as so secret that the expert can't see them.

So I want you over there on the Google side or the Oracle side, whoever has these, I'll order you both to get the stack of documents so we can deal with this at the end of the hearing. Understood?

MR. VAN NEST: We understand, your Honor.

MR. BICKS: Yes.

THE COURT: Okay. If you don't have them here, then you should -- at least Oracle should have them here because you're the one who wanted this hearing. So you better be prepared to hand them up.

Okay. Now, we come to the main event for today, which was to try to get me back into the swing of this case so that I can understand it. A lot of water under the bridge.

We're going to have the following things. We're going to have a 20-minute opening statement -- by the way, I want to just say one thing about the confidential documents.

We're going to have a public trial. Is there any member of the press here? Raise your hand if you're a member of the press?

(A hand was raised.)

THE COURT: Okay. One person anyway.

The public owns -- the United States of America and the public own this courtroom. It is not a wholly-owned subsidiary of Google or Oracle.

So when the trial comes, these documents that you think are privileged -- not privileged, but are so secret that they can't see the light of day, it would -- I can't say never, but possibly one, maybe, document in the whole case would ever be so important that we'd have to clear the courtroom.

So this miscellaneous stuff about proprietary information, especially if it's more than a year old, come on. Don't do that. We're not going to clear the courtroom over and over again. So just get with it. That's the way it's going to be.

If you want to start getting your emergency writ ready to take it to the Court of Appeals, God bless you. But this is a public courtroom and does not belong to the parties. And these people out there, the public, have a right to see what goes on in the United States District Court.

I don't know who is at fault on this problem about the documents, but in every case involving companies like yours, the parties overclassify and they don't deserve to be overclassified.

So maybe you're innocent here and I'm wrong, but we cannot run a public trial and constantly be excusing people from the courtroom. So be thinking about that.

All right. We're going to go to the opening statements and since Google has the burden of proof, I'm going to let Google go first. You get 20 minutes. Then we'll let Oracle have 20 minutes. And then we're going to go to the tutorial part.

Please go right ahead.

OPENING STATEMENT

MR. VAN NEST: Good morning, your Honor. Bob Van Nest for Google, and we appreciate an opportunity to address the Court and, I guess, our mock jurors as well.

This case is about Android and its phenomenal success in revolutionizing the market for mobile phones.

Android is the software operating system, the brains, if you will, that runs the most advanced and popular smartphones in the world.

Engineers at Google spent years and millions of dollars designing and building Android using Google technology and know-how and what they created was new, different, transformative from anything that had gone before it.

Google then published Android as open source software. That made it available to anyone wanting to use it to build a smartphone, a tablet or any other product. And so now we can get Android phones from Samsung, Motorola, LG, Verizon, Sprint and many, many others, and that's why Android phones are the number one selling smartphone in the world.

Now, we're here today because Oracle, which had absolutely nothing to do with the development of Android, wants credit for this remarkable success and is asking you, as jurors, to award billions of dollars in damages based on Google's success and Google's partners' success with Android. The evidence doesn't support such a claim.

Parts of Android were written using the Java programming language. Now, the Java language is free and open and always has been. No one has to pay to use the Java language. And there is no claim in this case that Android's use of the Java programming language was wrongful in any way.

The language was created by Sun back in the 90's and given to the world, made public, promoted for use in textbooks, universities and conferences all over the world.

Now, when programmers use the Java language, they also use a set of tools created by Sun and given to the world called Application Program Interfaces, APIs. And, of course, you're going to be hearing a lot about APIs during the course of this trial.

APIs are collections of functions, which we call methods, that a programmer might use to write a software program. An example of a method might be adding two numbers together, or comparing two numbers to see which is larger, or displaying a photo or an image on a screen. These are all methods within the Java APIs.

Now, the APIs are made up of two distinct elements. The declarations -- which are the labels given to the methods, and programmers use those labels to put the methods in their programs -- and the implementing code.

The implementing code is the computer source code that actually does the work of performing the function; adding the numbers, comparing the numbers, displaying an image or a photo on the screen.

The declarations, the labels, they make up a very small part of the Java Platform. Just based on code count, less than one half of 1 percent. It's the implementing code that does the work. The implementing code makes up the largest part of the Java Platform.

Now, as Judge Alsup told you, Oracle holds copyrights on the Java APIs. But Android is not using any of the implementing code in the APIs.

Google engineers wrote the implementing code for Android or they used other open source versions to complete the Android platform. There is no use of any Java implementing code in Android.

What Android does use are the API declarations, the labels. And these labels had been used for years, and they had been used for years when Google used them by developers, programmers, other companies, projects, open source; widely used and encouraged by Sun.

Now, the Java APIs at issue in this lawsuit, they were never, ever intended to be used in a smartphone. They were built for desktops, laptops and large servers. Devices that need a lot of power, have a lot of power, run at high speed.

Sun itself tried to adapt the Java APIs for use in a smartphone, but failed. Android was the first time that anyone anywhere had successfully modified, altered the Java APIs for use in a smartphone. So when it was announced, it was a phenomenon. And Sun itself recognized the revolutionary nature of the Android.

The CEO of Sun, who will testify during the course of the trial, said: Google, congratulations. We welcome Android to the Java community. We support Android's use of Java. We want to help because Android has strapped a set of rockets onto Java.

Not only that, but in the years that followed -- 2007, 2008, 2009 -- in public communications and private communications between the top brass at Sun and the top brass at Google Sun repeatedly said: We want to help you. We want to participate. How can we help? How can we be of assistance?

Now, why are we here? We're here because in 2010, years after Android was released and became successful, Oracle purchased Sun and acquired the Java programming language and wanted to use it for itself to build a smartphone.

First, Oracle tried to build what they called a Java

Phone. That project failed. They didn't have the manpower or the know-how to do it.

Then they tried to partner with Google. Mr. Ellison himself proposed a partnership with Google, but he had nothing to offer. So that failed.

So failing to build and failing to partner, they sued. They filed this lawsuit claiming for the first time that Android was an infringement that was unfair.

And not only that, but they are claiming billions of dollars in damages based on things like Google Search engine, which runs on Android, and the Google Ad products that run on Android. These are products that in and of themselves were phenomenal and they don't run any differently on Android than they do on a desktop or a laptop or any other device. And they certainly don't depend on the declarations in these 37 APIs.

So the question for you, as jurors, is whether Google's limited use of these Java API declarations, as part of something as new and different as Android, is a fair use under all the circumstances.

Now, Judge Alsup will give you a set of instructions on what "fair use" means, and he's in control of that. But in general, fair use is a use which was reasonable under the circumstances and, in particular, a use which was new, different and transformative.

If the use is a fair use, then the law allows that use

even without the permission of the copyright owner. There is no infringement if the use is fair. And in this case the evidence is simply overwhelming that Google's limited use of the Java APIs, the declarations, did create something new, different, transformative. It was praised by Sun. It was later praised by Oracle. And it was recognized around the world as something spectacular.

Now, I want to give you a little background in Android, because Android wasn't created by Google alone. It was created as part of an alliance with other companies. So Samsung, Verizon, Sprint, HTC, they all participated. The idea was to create a platform that everybody could use.

I actually have some slides, your Honor, that Ms. Anderson is going to use in the tutorial; but if I could hand them up, there is one slide that I'd like to discuss now, and that is the very last slide in this deck, Slide 5.

(Whereupon document was tendered to the Court.)

MR. VAN NEST: You may remember from our last trial the Android stack.

(Document displayed.)

MR. VAN NEST: The Android stack -- now, Ms. Anderson is going to deliver our tutorial and she will give you more detail about it, but the main point of it is just to remind you that Android is a lot of technology that has nothing to do with these Java API declarations. Right? It has applications,

framework.

All those libraries in green, those have nothing to do with the APIs. The APIs reside over there in Android runtime in the core libraries.

The point is, that these declarations were transformed themselves. All the implementing code was replaced. They were chosen because they are special for smartphone. They were integrated with 168 Android libraries -- the rest of them were also created by Google engineers -- and they were placed in this stack to create the first ever smartphone platform that was open and free for people to use. 15 million lines of code, that's up and down the stack. And Ms. Anderson will go into a little more detail.

So what did it create? It created a new platform, first, for people like Samsung, LG and so on. They had a new platform to build phones on. Then the carriers had a new platform that they could sell phones from.

And there is another group that this created add platform for, and that's the developers.

Now notably, your Honor, the developers don't have to write their -- you have a phone like mine. I've got it turned off because I'm trying to follow the rules in here, but my smartphone and your smartphone, they have lots of applications on them written by application developers, some are free and some are sold. They aren't all written in Java. The

developers don't have to write their applications in Java. They can write in something called native code and it runs just as well.

And, as a matter of fact, most of the popular applications on an Android phone are not written in Java at all. They are written in the native code or other -- or other codes.

So the developers, as a group, they follow the money. They go where the audience is. If you've got an iPhone with a great audience, they go there. The iPhone doesn't use any Java code. In you've got an Android with a big audience they go there. There is survey evidence that we will present, your Honor that proves that. They follow the audience, they don't care what the language is.

Now, by the way, Sun was always aware that Android intended to implement Java. As you recall, Google and Sun discussed a partnership. The idea was that Sun would contribute all of the Java technology. That means all the patents, the trademark, the Java cup. All the APIs including not just the declarations which everybody was using anyway, but the implementing code, which was understood to be protected. That was the idea. Google would put the know-how together. Sun would provide all this technology and they would call it a Java Phone.

Well, that never happened because the parties couldn't reach terms, and that's why my phone isn't called a Java phone.

It's called an Android phone. And that's why Google engineers spent several years and millions of dollars developing the phone because they didn't get all the technology that would have been necessary as part of their partnership.

Now, the Java language, as Mr. Ellison finally admitted last time, is free and open. Anybody can use the Java language without paying Sun or Oracle anything.

The Java APIs are used with the language. You can't do anything much useful without using the APIs. That's the whole point of using the Java programming language, is to have the shortcuts, the methods, the functions that help you write a code.

And, by the way, their witnesses concede that. We've got testimony from Mr. Smith, Donald Smith, a senior engineer at Oracle. He says:

"Yes, the APIs are a fundamental part of what makes Java Java. That's what makes a developer recognize Java. The APIs are a fundamental part."

And, as you know, Java was spread throughout the world. Everybody learned to program in Java. I think we learned your Honor learned to program in Java, but Java was taught all over the world and meant to be -- meant to be free and open, according to Sun.

How do we know that? Mr. Schwartz testified, and will testify, that the business plan was to get the APIs out there.

I asked him point black:

"QUESTION: So were the APIs simply marketed along with the language? In other words, free and available for everyone?

"ANSWER: Absolutely. We talked about open APIs. And then you compete on implementations" -- that's the implementing code -- "and what that means is we all had the same set of APIs."

So as a result of that, Sun actually encouraged companies to do exactly what Google has done here: Encouraged companies to develop alternative versions of Java using the same API declarations and writing your own implementing code.

The best example of that is a Apache Harmony. Apache Harmony took the same Java APIs, the declarations, wrote their own implementing code and made that available for people and companies to use.

Google used it in Android. IBM has used it in several of their products. Other companies use it. And Sun absolutely encouraged and endorsed it as part of the business plan.

More than that, shortly after Google launched Android, Sun decided it would make its entire API set, including the implementing code, free and open. In 2007 they announced what they call Open JDK. That's the Java Development Kit. What that encompassed was all of the Java APIs and all of the implementing code, given to the public for free to use.

Like many other Sun developments, they simply released it. It's open. Anybody can use it. It's like Apache. It's like the other projects. There no -- there is no payment required for any of that.

Now, I'll leave to Ms. Anderson the revisit to the file cabinet, but your Honor will remember the file cabinet. And the point of the file cabinet is simply that these API declarations that we're fighting about are little more than the labels, the addresses, of where the hard work of Android resides.

You can envision a file cabinet as a Java package, like java.lang. You can envision the drawers as the classes, like java.lang.math. You can envision inside the drawers, you can envision the -- I guess I've got them right here. No, I guess I've got them right here.

You can envision inside the drawers the file folders that have the name of the method. This name is what Android used (indicating).

But all of the source code, the implementing code was unique to Android. It was taken from other open source projects or written from scratch. And that's our point. This is what drives Android, not the labels.

And Ms. Anderson will go over that with you in a little more detail, but that was the -- that was the file cabinet.

So Google's use of the declarations in this Android stack

was a fair use and, as you know, in general you can consider all the circumstances surrounding the use as part of fair use, but the law does call out for -- in particular, for consideration.

The first one is whether the use is simply a copy or whether it's somehow transformative. And I think, as I've said, the evidence in this case is overwhelming that Android is not a copy of these declarations. It's a transformative platform that was new and different. Even a commercial use can be fair use if it's transformative use and the other requirements are met.

So what is the evidence? The evidence here is that even Sun thought this was a fair use. Sun welcomed Java -- excuse me, welcomed Android. Indicated it was a good thing. Offered to promote it. Did promote it. Never objected, but not -- beyond that, offered to promote and did promote it.

Now, why do we know it was transformative? Both Sun and Oracle tried to use the same APIs to build a smartphone and failed. Sun did it in 2008, 2009 with a project called SavaJE and a number of other projects which we'll visit during the trial, but they were all failures. And when Mr. Ellison got ahold of it, he tried, too. That was a failure as well. Why? Because these APIs were really developed for use in servers, laptops, desktops; never a smartphone.

And in many ways the enormous success of Android, which I

know you're going to hear a lot about from our colleagues at Oracle, both financially and in market share and in use around the world, that success itself is a testament to the transformative nature of Android. That's what transformative products do. They take off and drive new markets and attract new investment and create a new platform. And that's exactly what happened here.

The internal documents at Sun make this absolutely crystal clear. Not just the public pronouncements of Mr. Schwartz, but internally they said:

> "We are thrilled to have Google amplify the global momentum behind Java technology and deliver open source contributions of new services and new frameworks. We look forward to collaborating on the evolution of Java with Google."

I mean, that's the very definition of transformation.

And Mr. Ellison, too, recognized this. You remember the video of Mr. Ellison at JavaOne:

> "We're flattered. Flattered by Android's use of Java and we expect to see lots of Java devices from our friends at Google."

Their own witnesses now admit that Android is transformative. The testimony from Terrence Barr, one of their -- I think he was actually a 30(b)6:

> **"QUESTION:** Was Android transformative, Mr. Barr?

"**ANSWER:** I think I already answered that, that I believe in some ways yes."

Mr. Barr says Android transformed telecommunications industry itself. And we'll present lots of evidence about transformation.

The three other factors, I think, are a little -- the next two are pretty straightforward. The nature of the work. Is it creative work? Is it functional? Creative work: Is it a book? Is it a movie? Is it a play? Or is it a functional thing? Obviously, Android is a functional thing.

Mr. Ellison called it "a command structure that helps operate the program you're using." So, clearly, clearly Android is not a novel or a play. It's a functional device.

The amount and substantiality of the copyrighted work used. Very important to remember, your Honor, the copyrights at issue here are 1.5 SE, Java SE, Standard Edition 1.5 and 5.0. They may get up here and say: Oh, Java was all over feature phones. The old flip phones. Well, that wasn't Java SE that was used in feature phones, and they were only feature phones. It was Java ME, which isn't even at issue in this lawsuit.

But as to what Android used from Java, it's obviously very small. The declarations, as I already said, make up less than one half of 1 percent of the Java Platform, which has something like 3 million lines of code. The declarations, even if you

count up every single line and word, amount to about 7,000 lines out of that 3 million.

The final factor is whether or not the use has hurt the market for the copyrighted work. So we're talking now about Java SE 1.5 and 5. And the evidence is going to show that Android has helped Java SE, not hurt it, but that Oracle has been unable to take advantage of that.

What do I mean by that? Just last year, just last year one of the leading industry groups that follows this says:

"Java regains spot as most popular language in the developer index. Java's rise is not seen as a one-time anomaly. It's not a fluke. Java is really catching up" -- to other languages, they mean -- "because of Android's continuing success."

Right? Android was out there. Android created a market. Android created an audience and so more developers chose to wrote in Java -- chose to write in Java. Why? Because they want to reach the audience. Number one.

Number two. Oracle has failed to capitalize that because their own people admit they failed to invest in Java. They failed to invest in Java.

Now, when they got Java from Sun, it was already in decline.

The Sun documents indicated, and I'll quote a couple that we looked at last time:

"Sun's leadership" --

THE COURT: All right. Take a minute to -- you're over the 20 minutes. So take a minute to conclude.

MR. VAN NEST: Okay, I'll do it.

The point is that Oracle and Sun recognized that Java was stagnant. Java was legacy. Java was not changing. They weren't doing anything to make Java as good as it could be.

And they made a decision internally at Oracle not to invest in smartphones with Java. They made a decision back in 2011 that it wasn't worth their time and now they are complaining that they lost profits. Well, any profit they've lost is because they didn't invest and keep Java current and allow it to grow as it should have had they tended to it.

So my final point, your Honor, is that Oracle doesn't deserve any credit for the success of Android. Android is successful because of Google technology, the Google brand, the phones that partners build, the developers' commitment to the platform and all of that other stuff. Right?

Oracle claims that Java APIs attracted developers. Well, in fact, it's the other way around. Android attracted developers.

Ask yourself this question: How could it possibly be that these declarations are the cause of success?

iPhone, which is the other smartphone that's very successful, uses no Java whatsoever. There is no Java in an

iPhone and, yet, it's a very succeed smartphone.

The experts in the Java API, Sun and Oracle, they both tried to use these APIs to build a smartphone. They failed.

So it can't possibly be that all the money that's been generated for many people, including for Google, based on Android, has been created by the use of these 37 API declarations and labels. Common sense tells you that.

The other thing that tells you that is their big ticket items are all things that Google developed completely of Android. Their big ticket money items are the Google Search revenues and Google Ad revenues.

Well, Google Search and Google Ad are themselves phenomenal products with phenomenal technology that everybody recognizes were groundbreaking. They don't run on Android any differently than they run a desktop or a laptop. It's the same product and they're trying to take credit in their damages demand for all the hard work of lots of other Google engineers far beyond Android when, in fact, the Java APIs have nothing to do with the success of Google Search and Google Ads on the Android platform.

So that's what I have, your Honor.

THE COURT: All right.

MR. VAN NEST: And I thank the Court for giving me an opportunity to address you.

THE COURT: All right. Thank you.

So who is going to give the counter argument?

MR. BICKS: I am, your Honor.

Would it be possible to turn on the screen here?

Your Honor, I've got a few graphics, if I could hand up, that are going to --

THE COURT: Well, you're limited to five total.

MR. BICKS: Understood.

(Whereupon documents were tendered to the Court.)

OPENING STATEMENT

MR. BICKS: Good morning, your Honor, and may it please the Court. On behalf of Oracle, my name is Peter Bicks and, as the Court requested, for the next 20 minutes I'm happy and proud to give you a mini opening of our case.

My partner, Annette Hurst, will then present the technology tutorial when I'm done.

What is the case about at the end of the day? Why are we here? The case is, in a nutshell, about a business decision that was made 10 years ago by Google to take, without permission, Oracle's valuable property. Property that they took at the time knowing that they needed permission.

The property that was taken was computer software. The property is protected under the copyright laws of the United States. Laws so important that they are written into the Constitution and the purpose of which is to protect and encourage people who invest their resources, their heart and

their soul, like the people at Oracle did.

The Court has instructed the jury in this case that Google's taking infringed Oracle's copyrights and that Google can only avoid the consequences of its decisions if it can prove that taking Oracle's valuable property was fair use.

The evidence will show that Google's conduct over the past 10 years was the opposite of fair use and that Google chose to engage in a course of conduct that is in clear violation of the law.

The evidence will show that the copying here is of monumental proportions, and it's copying that continues today.

Oracle's property is in every mobile phone known to the world as Android that has been sold for the last eight years and is selling today. In total, almost 4 million -- billion, billion mobile phones and other devices have Oracle's property in them, all without permission.

Google's huge gains with Android were Oracle's loss. We have 10 years' worth of evidence, much of it coming over the last five years, showing how Google's unauthorized copying harmed Oracle. And it's just not actual harm because, as we'll discuss and the jury will be instructed, it's potential harm as well under the fourth factor of fair use.

The evidence will show that Google has made billions of dollars of revenue, billions of dollars of profit, all at Oracle's loss.

So let me start with my first graphic. I have a chart there that -- is there a way that I can see it on my screen here?

(Brief pause.)

(Document displayed)

MR. BICKS: So, your Honor, I now have a graphic that kind of shows the mobile phone market share.

And I'm starting out here. You can see, you've got Java on the top and you've got Android on the bottom.

And if I can click this thing, which I'm trying to do...

Dawn, I don't know --

THE CLERK: You've got to turn towards probably your PC or something to get your -- I don't know.

(Brief pause.)

MR. BICKS: Here we go.

(Document displayed.)

So what I have done here, your Honor, is I have shown 2007. You see Java at the top there. Java was a little under 80 percent market share in mobile phones and had about -- in 1.8 billion phones at that time.

Android hadn't even started. They were ceding the market in 2008. 2007, 2008 they come out with their first Android phone, and you can see the pattern. Java is going down, Android is coming up.

And I intersect there at around 2010, 2011 because in this

case that's the first time around. That's when kind of discovery stopped.

So what -- what's happened since then?

(Document displayed)

This is what happened since. Android has taken off. Java is down to nothing. That's really what the case is about at the end of the day.

And then what we do is we question whether or not Google -- Google's copying is fair use. And what's the evidence going to show? There is nothing fair about what Google has done. So let's look at what the fair use factors are.

Here they are. There are four of them.

(Document displayed)

They've got the burden of proof on each of the factors. They will fail to meet their burden of proof on fair use.

Factor One: Purpose and character. The evidence will show in this case that Google's copying of the Java API is commercial and is not transformative.

Factor Two: The nature of the copyrighted work, the Java API. It's highly creative.

Factor Three: They took the most more of the Java API. There is a reason they took what they took.

And Factor Four: The potential market, probably the most important factor, has been harmed. The actual -- and if this

conduct becomes even more widespread than it is, which is the question, it will cause serious harm, even more harm that's already been caused.

And those are the four factors.

So in this case we put it to Google: Is it commercial or isn't it? Well, maybe it is, maybe it isn't.

But I want your Honor to hear a clip where this question was asked to Google's counsel. The question was posed to Google's counsel:

"For purpose and character" -- which is Factor One -- "you don't dispute that it was entirely a commercial purpose?"

And the answer was:

"It's entirely a commercial purpose."

So I want you to hear this.

(Audiotape played in open court.)

So we look at Factor One. We have got no, no, no. There is no question that this is entirely a commercial purpose. Why? Why did Google take Java? They needed to act quickly. The Java was dominating the wireless industry at the time period that we're talking about, as that first graphic shows.

Java was in 1 billion phones in 2005 and 1.8 billion in 2007. And that was the year that Apple came out with the iPhone. So the momentum was looking pretty good for Java, but Apple was now on the scene.

Six million software developers were writing in Java. They were part of the Java community and they protected Java, along with Sun and then Oracle.

Java was great for writing computer apps. That's why people used it. That's why they loved it.

All of this, if Google could get ahold of it, would allow them to be quick to the market and meet the mobile window that they saw about to close.

Google was out promising phone manufacturers that Android would have Java in it. It was a critical selling point when they were out there on the street trying to get business partners.

But there was a catch. There was a catch. Google knew that those APIs were protected property under the copyright laws of the United States. And they knew they needed written permission in the form of a license.

After all, the other commercial partners -- most, if not all of them -- had written permission and worked out business deals to use that property.

Google first tried to negotiate, but the two sides couldn't reach an agreement because Google wanted to control. Because control is one of the drivers of economic success.

So they had a choice to make. Fork in the road.

One, am I going to invent my own property? Google certainly with all the resources and expertise one would have

thought they could have come up with their own technology.

Or, two, copy Java.

Google decided to copy Oracle's property because Google had to make sure that mobile window didn't close and they weren't left behind. They knew the entire time that the APIs were protected and copyrighted and that copying without a license was not right. And the copying continues today.

What did they copy? What did they copy? Ms. Hurst in the tutorial will go into that. But they copied over 10,000 lines of computer code, declaring code, from 37 what are called API packages. So they copy it once, but when they sell 4 billion phones, those 10,000 lines are copied 4 billion times.

And they also copied what was the blueprint for the hugely valuable created -- creative pre-packaged packages the programmers loved. Packages that took years to create in many instances and huge investments to create. And the programmers loved them. They made their job easier. They were intuitive. People didn't have to go back to square one. They could use these things and they were really, really helpful.

So, transformative. What did they do? What does transformative mean?

Google took Oracle's copyrighted code and put it in Android and they included it in new versions of Android released in 2012, 2013, 2014 and 2015. The code that Google took does the exact same thing in Android that it was doing in

the Java-based phones.

And in those situations Sun had commercial contracts until Android came along. And it was doing the same thing in Android that it was doing in those other devices. Not just phone, but things like touch screens in cars, tablets, eReaders, Kindle, printer and other things like that doing the exact same thing.

The code's use in Android is not transformative from a basic business perspective. Why? Something that is a substitute is not transformative. Android is a substitute for Java in the market. It is not transformative.

The second factor here, creativity. The evidence in the case from witnesses at Google themselves will be that these APIs, some of which took years to come up with, are creative. One of their experts compared it to being an artist, football player, violinist. There is no question that there is creativity involved in coming up with these APIs.

Factor Three -- and it will come up more in the tutorial -- you'll hear how important these packages are because you have to ask yourself a common sense question: Why did they take them if they really didn't need them, if they weren't important? They took them because they were important. And we'll learn in the case how important they were.

The classes that actually Google copied out of those packages are by far and away the most important classes. And what they copied to things like apps and computer programs,

stability is important. It's got to work right. And they copied parts of the code that made things stable. And, of course, when you remove it, what they copied, Android wouldn't work. Phones wouldn't even work.

So what was the strategy? Talking Factor Four, the harm, what at least the Federal Circuit here has said is the most important factor. Google's plan was pretty simple. Give away Android for free. Get as many users as possible and make money off of advertising. And I showed the Court the impact on the phone market.

Android I think, as counsel acknowledged, dominates the market. They've got over 80 percent of market share. Over 1 billion Android based smartphones were shipped in 2015. 1 billion. There are more than a million apps in Google's app store. The key ones, the key ones, as Ms. Hurst will explain in the tutorial, require critical parts of these APIs that were taken.

And there is no question from their own executives that Android is hugely profitable. Their words, not mine. And that goes back to Factor One: Is this commercial? Not only is it commercial, it's hugely commercial.

So what's happened since discovery stopped in this case?

(Document displayed)

The numbers to the right, because of confidentiality concerns of Google's, are not to the dollar, but they are

ballpark good. They are 10s of billions. Ad revenue right up there at the top. Revenue from apps. Hardware and digital content.

Last time we were here it was about 880 million. Now, tens of billions of dollars.

Java's market share has plummeted, as have the number of the Java based smartphone sales, and the number of developers who are using Java has also dropped dramatically from internal documents. Salesmen and women out in the field trying to sell, Android has severely damaged handset Java sales. We are getting run over in the market. I see Android in the market.

And we're not just talking about potential harm in this case. And this is important. We're talking about actual harm. We're talking about actual lost mobile phone licensing business: Samsung, Motorola, LG, HTC, Sony, Sprint, Verizon, AT&T.

As one individual wrote:

> "I see Android and I'm run over by it in all of my accounts."

Actual harm under Factor Four, which, of course, goes directly to the question: What are the damages in the case?

And there's -- in addition to actual observed harm in the form of deals lost, we've got to look at potential markets, because that's so critical under Factor Four. Potential markets. What if everybody did it? What if everybody could

come take Oracle's copyrighted property? And that's going to be the question under what is the most important factor in the case, potential harm, beyond the actual harm in the case.

And what is quite astonishing about what's happened here is these revenues, many of them are done with business partners, former business partners of Oracle's, the same partners who were working with Oracle when Oracle had that close to 80 percent share in the mobile phone market.

So we've got to ask the question: What is the harm to Oracle if all market participants could be allowed to copy the Java API without any kind of a license, without written permission?

As I alluded to before, we've got to look at potential harm to both the Java Platform and products that use that platform, what would be called derivatives, because that's the inquiry under Factor Four. It's potential markets.

And it's very clear that Java is in multiple markets. It's in smartphones, eReaders, tablets, and all of the other things that Android is in as well. Java was in all those markets all along, and now Android has gone head-to-head in those markets with the code that was taken in this case. All of the new versions that have come out, and the Court has instructed the jury in this case contain our client's property, are now in multiple products, multiple markets going head-to-head.

And for Factor Four, all we would need to have is one lost deal and that would show harm, but there are multiple areas of actual and multiple areas of potential harm.

So we ask ourselves: What if everybody could do what Google did? What if everybody could do an end-run around the copyright laws and an end run around somebody's business strategy to do well with property that they own, that they bought, and that they invest in? And the answer is: It would run roughshod over the company. More harm than we saw in lost deals if everybody could come and just start taking their property. People who have existing deals, who negotiated contracts, like business people do, would come back to the table and say: This is free now. I don't have to do a deal with you.

And there would be broader problems even beyond that because there would be no incentive for companies like Oracle, who buy assets and invest in them, to continue to do it if people could just come and take that property without permission, take their business partners and do deals and push them out of the market.

So at the end of the day the question is going to be: Is that fair?

Google will not meet its burden. And we could stand and say: You've got to meet your burden, which they do, but we would present our evidence to show not only that this is not

fair, but it wasn't done in good faith. It was done with complete knowledge that a license had to be entered into and it was done with complete knowledge, written by people who were there at the time. Before we would come into court and lawyers could look at documents, they put in their own documents that they knew they would make enemies along the way by doing and embarking on the course of conduct that they embarked on. And they did. And that's why we're here.

Thank you, your Honor.

THE COURT: Thank you. So let's -- let's turn to the tutorial part. I guess we'll go back over here.

What did I say 20 minutes? How much time did I give you for this?

MS. ANDERSON: Your Honor, I don't believe that you specified, but I can work to fit it within 20.

THE COURT: Let's shoot for 20 minutes on each side on the tutorial.

MS. ANDERSON: Can I grab a water?

(Brief pause.)

MS. ANDERSON: Good morning, your Honor. Christa Anderson on behalf of Google.

Thank you for giving an opportunity to give a brief tutorial today to address some of the basic technology issues that we think may be helpful for the Court to understand in embarking on our retrial.

The matters that I will be focusing on today will be in four areas.

First, I'm going to discuss a little bit some concepts concerning Java: Java the language, Java the platforms and Java APIs and their SSO and declaring code.

I'm also going to touch on the Android open source platform.

I'm going to discuss certain concepts relevant to open source licenses, which is licenses applicable to some of the technologies you'll be hearing about.

And, finally, I'm just going to touch on a few technologies that you may hear about in the course of the case that will help the Court better understand the parties' positions on certain matters.

So let me begin with an overview of Java. The term "Java" is used in different ways depending on the context. So I'm going to discuss some of the contexts that you'll probably hear the most during the course of this case, your Honor.

First and foremost, Java is often used to refer to the free Java programming language. Programming language, as the Court is aware, allow programmers to express ideas in words that ultimately get executed by a machine in machine code.

These languages, they can differ, as your Honor is aware, in grammar and syntax, but often high level concepts are expressed in similar ways. For example, a Java language

programmer probably could understand the general gist of a program written in the C++ program and vice-versa.

In fact, similarities between Java and other languages are often taken advantage of. Oracle itself in its own publications acknowledges the fact that the Java language purposely derives much of its character from C and C++ because it helps with understanding by computer programmers.

Examples of these similarities include an API name called RegExp, which for regular expressions, which is a concept that has been around for decades and is incorporated in Java, as in other prior languages.

Similarly, words like "Bool" for Boolean or "Char" for character, we see those over and over again going back to C and perhaps even earlier because, again, they take advantage of these similarities in Java and other languages to make communication easier.

Now, necessary for the use of languages like Java are something called APIs. Much like the word Java, the word -- the term "APIs" can be used to mean different things, depending on the context. So it is sometimes, but not always, used to refer to the label associated with a particular set of implementing code that actually performs the function being called up.

In the world of programmers, these API labels really function like a set of vocabulary and grammar that allows the

programs to communicate with one another. So, for example, one program can use an API label to ask a different program to do something for it without needing to know anything at all about how that second program actually goes about doing it or implementing it.

So this is sort of a helpful feature of APIs. It allows communication without needing to know implementing code by the second program.

Now, the API labels because of this are really just an abstraction. They, themselves, are not the code that performs the function that's being called up.

Control+P is a common example that could be used to explain this idea of an API label being an abstraction. That Control+P function is used in many different situations to call up the function of printing something without necessarily knowing how the machine you're using it on is going to do that. The implementing code may not be visible to you, but using this label Control+P gets it to happen.

Now, APIs consist of various methods that for certain programming languages get grouped. Methods in an API get grouped within a class and classes can get grouped within packages.

THE COURT: Can you help me on that part?

MS. ANDERSON: Yes.

THE COURT: I want to have the exact -- the sequence

in descending order. And you were right on this point, so help me. And where does the word "library" fit in there?

So go through what you just described; methods, packages.

MS. ANDERSON: Yes. All right. So, and it may be helpful to the Court, if your Honor would like, I also have for you a set of graphics.

Slide one, which is a visual that also you could write on, if that's helpful to you.

THE COURT: Let's see that.

(Whereupon documents were tendered to the Court.)

THE COURT: Okay. Cartoons always help me, but...

MS. ANDERSON: We like them, too.

THE COURT: All right. So -- but don't see -- all right. Explain this to me and maybe it will become clear.

(Document displayed.)

MS. ANDERSON: Let's talk about it this way. So, again, this is the file cabinet analogy your Honor probably remembers from trial one. And we're talking about a method called max.

To call up max, the programmer is going to want to know what package it's in, what class it's in, and the name of the method. So written at the top right there java.lang.math.max, that actually tells you it is in a package called java.lang. And that is depicted by the file cabinet, the lang file cabinet you see there. It is in a drawer, the class math. So the

drawer symbolizes the idea of a class, which is a subpart within the package. And then the five --

THE COURT: All right. The method is called max, right?

MS. ANDERSON: Yes.

THE COURT: Let me write that. And math is called what?

MS. ANDERSON: Math is the class.

THE COURT: So what is lang called?

MS. ANDERSON: That's the package. We call it java.lang. We get that package, which is the file cabinet.

THE COURT: Where does the word "library" come in? Remind me about that.

MS. ANDERSON: Well, often that is a concept. Again, it can be used in different ways depending on context.

But when you're talking about libraries, for example, the core libraries in Android, you're talking about a collection of implementations for various methods which are then grouped within classes and within packages.

And so libraries, sometimes you hear a reference to core libraries, that's a reference to this collection of APIs which incorporates two parts, declaring code and implementing code.

THE COURT: But the thing java.lang.math.max is that the declaring code? It will come back to me, but I can't remember now.

MS. ANDERSON: Well, you'll see in it a second, but, yes, that's a label. And that label is reflected again within the implementation, and there it is called declaring code or declarations or method headers. And we're going to talk about that in, I think, a more helpful graphic, the next slide.

But that is the concept, is these are nested within each other. And programmers who are programming need to know what package it's in, which class it's in, and what the name of the method is.

Now, those are all labeling devices though, your Honor. Again, the API being used as a label. But where is the code that actually does max? Where is the code that actually performs the function? That's within the -- this depicted file folder, the thing that says "max" there, the manila folder. Within that is actual -- an actual implementation that has implementing code in it.

So let's take a look at that. If your Honor could turn to Slide 2?

THE COURT: Okay. I'm looking at it.

MS. ANDERSON: All right. So here we have depicted for your Honor an example.

On the left-hand side you see just the simple name of the method "max" that provides the greater of two numbers. But what is the implementation? Well, that's on the right. And here we have the implementation of max in Android. And this

implementation, as Mr. Van Nest explained, contains two parts.

At the top you're going to have something called a method header. Also called declaring code. Also called declarations. That's the line that says: Public static float max of float F1, float F2.

And then below that, all the lines below it, well, that's the actual implementing code. That's the part that tells the computer how to do it. That is not accused. That's the bulk of all of the code that you're going to find in these libraries and it's not accused of infringement in this case.

What's at issue are these method headers, also called declaring code; that first line that basically just restates the label, the label for max.

THE COURT: All right. Stick with the source code part for a minute.

MS. ANDERSON: Sure.

THE COURT: The word "public."

MS. ANDERSON: Yes.

THE COURT: I used to know the answer. What does that mean? Remind me what that means and what "static" means.

MS. ANDERSON: All right. Okay. So the reference to the word "public" is that public methods are part of a class' public API that the developers can use. When developers are writing program, they can use that.

"Private" is a different kind of modifier that

indicates -- private methods are things that can't be called by programmers writing programs using a class of APIs, but, rather, they can only be called from within the class internally as part of implementation.

So here we have a public method because it's something developers can use. The word "max," the method max they can invoke.

THE COURT: So the parenthesis and max, you don't have it in there, but it would actually include F1 and F2 within the parenthesis; is that true?

MS. ANDERSON: Yes.

THE COURT: Is that the way it works for the actual programmer, so that --

MS. ANDERSON: Right. So the method -- the method header, the declaration, whatever you want to call it here, declaring code, it's all the same, consists of three things: The name, the arguments and the return.

The name is max. The arguments, which is what the method expects to receive when it's invoked, are in parentheses, float F2 and float F2 --

THE COURT: But over here you didn't put F1 and F2.

MS. ANDERSON: No, I did not. It was just --

THE COURT: But it would really -- in real life it would be in there.

MS. ANDERSON: There would be an indication of

arguments in there, correct, your Honor.

THE COURT: All right.

MS. ANDERSON: And then -- then the return you see here indicated as float, that's the float type. The return type is float, a number with a decimal point.

So that's just kind of basic information labeling the method, flagging the name, the arguments and the return. Those are the method headers.

And the reference to static, your Honor asked, is just a reference to the type of variable involved.

THE COURT: All right. Can we go back to the first page a minute?

MS. ANDERSON: Sure.

THE COURT: And maybe Oracle will help me on this, too, when it's their turn. I may not have this right, so I want to make sure we have the issue right.

The Court of Appeals disagreed with me. I had said that the SSO was not copyrightable. That was a method under 103, I believe or -- is it 103 or 102?

MS. ANDERSON: Correct.

THE COURT: Anyway, but they said it was copyrightable. But, nevertheless, we're dealing with the Structure, Sequence and Organization, not the actual implementing code. I think everyone agrees that's been done by Google or gotten from someplace else.

So focus just on the SSO. So you have it here java.lang.math.max.

MS. ANDERSON: Uh-huh.

THE COURT: So if you had just made one change, let's say. Let's say instead of using l-a-n-g, you -- or let's say you did use lang, l-a-n-g. Let's take it under math. Let's say instead of "math" you used "arithmetic" and let's say that's the only change you made.

MS. ANDERSON: Okay.

THE COURT: Would that be an SSO violation?

MS. ANDERSON: I may not be tracking the ultimate point of your Honor's question.

I mean, the SSO -- my understanding of what Oracle's claim is that they are claiming the use of this -- this labeling process, organization that involves --

THE COURT: See, what I'm trying to get at is how -- normally, normally in a copyright case -- let's just take an ordinary. Let's say you've got a newspaper article and somebody goes along and they change enough words that it's a little different, but it -- all right. That's not a copyright violation. That's a -- because it has to be the exact words, or very close to it anyway. At least that's what I understand the law to be.

So I'm asking if that applies here, and what is it that we're fighting over? And so let's say that instead of using

the exact sequence and organization that Java uses -- which, apparently, on these 37 that's what you did do -- what if you had rearranged the file cabinets a little bit, it would have been confusing maybe to the users, but it would be -- would it be different enough with just a slight change that the copyright laws would no longer find infringement? That's what I'm asking. Because we're only dealing with SSO. We're not dealing with implementing code.

MS. ANDERSON: Your Honor, I don't know what Oracle's position would be on --

THE COURT: I'm asking your position. Come on. Don't do that to me.

MS. ANDERSON: No, no, no. I understand. I --

THE COURT: You give me the answer. I don't care what they say.

MS. ANDERSON: Okay. Well, I don't think any of it should be -- is constituting infringement because fair use --

THE COURT: Give me the decision on point that helps me with that.

MS. ANDERSON: Yes, your Honor, indeed. Fair use --

THE COURT: No, no. We're not talking fair use now. I want to understand what it is that we tell the jury that constitutes the infringement.

The jury -- listen, the jury is going to get in the room and say this: Listen, all they had to do was rearrange these

file cabinets a little bit and it would not have even infringed to begin with. So that must be fair use because we're dealing with something that's so -- this organization can't be that important. This SSO can't be that important. They might say that.

But then somebody is going to say: Well, maybe that would have infringed anyway. So the power of the SSO is much more important than just rearranging the file cabinets a little bit.

But maybe that's not right because copyright -- copying is copying. It's not -- see what I'm saying?

MS. ANDERSON: I see.

THE COURT: So I don't want to know what their argument is. This is the time for you to stand and deliver and give me what the law is.

MS. ANDERSON: Yeah. Well, your Honor, certainly when you're talking about what has been done, this particular java.lang.math.max, the reason it's written this way is because it needs to be used this way to use the Java language. And that's something Oracle witnesses acknowledge.

These APIs, which is the labeling, the taxonomy, the SSO, whatever you want to call it, the reason it's done that way is because you need to do it to be able to --

THE COURT: To be able to use those 37 APIs?

MS. ANDERSON: To -- to -- it is part of the reasonable use of this language, which Oracle acknowledges is

free. And that is why you see multiple independent implementations of these APIs -- not just by Google, but by others before we ever released Android -- because these are the kinds of things you see over and over again in the history of computer usage because that's how -- that is how computer program is done. If you -- but if you switch --

THE COURT: All right. You are not answering my question. I'm taking up your time. I don't want to do that.

MS. ANDERSON: Okay. Okay, your Honor.

THE COURT: Maybe Oracle can answer that question.

MS. ANDERSON: Thank you, your Honor.

THE COURT: All right. Go ahead.

I want you to answer that before we get to the trial. I want to know the answer to that question.

MS. ANDERSON: Happy to explore it for your Honor, yes.

THE COURT: All right.

MS. ANDERSON: Let me turn now quickly to the subject of Java platforms, your Honor, because maybe that will help answer these questions in some fashions.

Distinct --

THE COURT: Okay. I have a maybe easier question: You say that -- okay. Taking these APIs, the 37 and this SSO for the 37, you concede that you did that. You can't get out of that. And you -- apparently, you conceded to the Federal

Circuit judge that it was entirely, was the word, commercial.

So under the Federal Circuit's Rule, does -- if it's entirely commercial, does that destroy you, your position on whether or not it can be a transformative use?

MS. ANDERSON: Well, your Honor, I haven't heard the whole transcript of that hearing, so I cannot speak to the full context of what happened there.

I can tell you that, obviously, Android is used in many non-commercial ways all the time. Android is offered for free. Google doesn't sell Android. It is offered for free for use for anyone.

So -- so the question at hand here in terms of that first factor, it isn't just: Is it or is it not commercial? The question also is: Is it or is it not transformative and how does that weigh with all the factors?

THE COURT: What did the Federal Circuit say in its opinion on transformative and commercial?

Oracle told me a minute ago that if it's commercial, it cannot possibly be transformative. That's what I got out of their opening statement. Is that -- is that the law?

MS. ANDERSON: That is not the law, your Honor. That is not the law.

THE COURT: Then you all are going to have to help me understand what the Federal Circuit said on that point.

MS. ANDERSON: Well, and I would be happy to look it

up for you. Right now I have it behind me.

But the bottom line here is when it comes to that first factor -- and remember, all these factors, they are not exhaustive. These are factors that under -- for example, the model jury instructions for this, for this circuit, when you're looking at copyright, when you're looking at fair use, is it reasonable under the circumstances at hand? And here are the factors that you should consider --

THE COURT: Listen, we have some law of the case here. We have the Federal Circuit weighing in on this in a case that you all had.

I'm not saying I wouldn't vary a little bit from what they said, but basically I'm just going to lift what the Federal Circuit told us and tell the jury that. I don't see how I can possibly go wrong with that.

If I start going off into some lawyer-prepared instruction that the Federal Circuit has never vetted, then I may get myself in trouble. I will listen to it. Maybe you're right, but I need to stick pretty close to what the Federal Circuit said.

MS. ANDERSON: Understood.

Well, your Honor, when it comes to the first factor, whether or not the use is commercial, of course, is part of what is weighed; but, also, part of what is weighed is is it transformative?

And it is not the law that if you are commercial, you can't be a fair use or you couldn't be transformative. There is no --

THE COURT: Is that what the Federal Circuit said or that's what somebody else said?

MS. ANDERSON: The Federal Circuit, I don't think, laid that out in that fashion, but let me pull it up for you, your Honor.

THE COURT: Would you?

(Brief pause.)

MS. ANDERSON: Your Honor, I have a copy of the slip opinion that talks about the first factor and the fair use inquiry involving purpose and character of the use, including whether such use is of a commercial nature.

Then the Court says:

"This factor involves two subissues. The first is whether and to what extent the new work is transformative. And the second is whether the use serves a commercial purpose. Those are things that are considered separately."

The Court goes on to explain that use is transformative if it adds something new.

It talks about the critical question being whether the new work merely supersedes the objects of the original creation or instead adds something new and explains more about what

"transformation" means.

It talks about the fact that Courts have described new works as transformative when the work uses copyrighted material for purposes distinct from the purpose of the original material and cites a bunch of cases to talk about that, when uses are transformative.

It says:

"A work isn't transformative where the user makes no alteration to the expressive content or message of the original work," and talks about cases that support that.

And it says:

"The analysis of the first factor requires inquiry into the commercial nature of the use. Use of the copyrighted work that's commercial tends to weigh against a finding of fair use."

It isn't conclusive.

THE COURT: All right. I remember that.

So maybe I misunderstood Oracle, but it is not the law that commercial rules out transformative. Do you agree with that or not?

MR. BICKS: I actually didn't mean to suggest it was that clear, Judge.

What I'm saying is essentially the commercial nature eclipses in Factor One any argument about transformative

because you have to look at everything kind of in degree. That was really my point.

THE COURT: All right. Well, what I'm getting out of this is commercial tends to undercut transformative, but does not rule it out.

MS. ANDERSON: Correct, your Honor.

THE COURT: That's what I get out of what you read to me.

Okay, that helps me.

MS. ANDERSON: Okay.

THE COURT: All right. Now, I got you off of the --

MS. ANDERSON: That's okay. I can -- let me breeze through a few concepts. I won't take too much of your Honor's time, but I want to just touch on a few things that may be helpful.

How much time do I have?

THE COURT: You're getting close to the end. Let's see. I think you're already over, but go ahead. Take two more minutes and then we'll go to the other side.

MS. ANDERSON: All right. I just want to touch on something I think is important to understand.

Java platforms. Sun Oracle has offered three Java platforms that are sort of going to be relevant in terms of hearing the evidence here.

Java SE, obviously, is the platform related to the

copyrighted work at issue before your Honor. That's a platform that was targeted at server and desktop systems. The way that it was set up, its virtual machine, the software, the APIs are targeted at server and desktop environments that have more power, memory, clock speed. This is not a platform. That is not Oracle ever targeted like smartphones, much less a full stack operating system.

There is only a small part, a tiny part of the Java SE platform that's at issue in this case because that platform includes, again, software for all sorts of functionality: Virtual machine; APIs that have nothing at issue in this case; implementing code, again, not accused in this case.

The only parts at issue are the SSO and method headers, which your Honor might see as a helpful depiction in the graphic you have before you in Slide 4, which is just -- again, in the background you see the picture of that Android stack, which we've talked about with your Honor many times at Slide 3. But it's just a tiny part of this platform that's been accused. And that -- that particular SSO and the method headers for Java SE, tiny part of the Java SE platform.

Java EE, in contrast, was something that was targeted at server environments, had many more APIs.

And finally --

THE COURT: Okay. But if you make that argument, if it's such a tiny thing, why did you need to even use the SSO to

begin with?

MS. ANDERSON: Because again, your Honor, this SSO terminology, which we are using and have been using, the Federal Circuit called it taxonomy and we call it labels.

But the bottom line is Sun/Oracle teaches the world to use the free Java language and teaches them that to use it, you need to use these labels, this grammar approach, which is how we named these particular API labels for Android, because they are part and parcel and needed to use the Java language in this way in a way that Sun/Oracle applauded and encouraged for years, before and after we released Android.

So for Sun/Oracle to now claim that somehow, although it had encouraged folks to use it for free for years, after Android is successful to try to push the toothpaste back in the tube, that is not a -- not fair of Oracle. It's absolutely a fair use for Google to have used Android the way it did and implement what it did the way it did.

THE COURT: Is there anyone else out there who has used Java in some analogous way that -- without a license?

MS. ANDERSON: Well, remember, your Honor, this concept of Open JDK -- and I know you don't have a lot of time. But Open JDK is an implementation of Java SE that Sun/Oracle gave to the world for free under an Open Source license.

THE COURT: Well, what's the difference between Open JDK and regular Java?

MS. ANDERSON: Open JDK is an implementation, a different implementation of the same SSO containing the same method headers that are at issue in this case.

THE COURT: What's the difference then?

MS. ANDERSON: There is none for purposes of your question in terms of what --

THE COURT: Are you saying the SSO is identical? Is that correct?

MS. ANDERSON: For the 37 Java APIs, yes.

THE COURT: All right. But then it must be different in some way. What is the difference?

MS. ANDERSON: Well, I couldn't catalog all of them, but among them include the fact that the implementing code that is contained in Open JDK, I expect, differs from the implementing code in the Java SE platform in certain ways.

There are other differences that may not be material to this litigation.

But the reality is Sun made a decision in 2007 that it benefited the Java SE and other platforms to make sure that this was out there in an Open Source version, to make it popular, to reinvigorate the Java language. It knew to do it, it had to get on the bandwagon of Open Source. It made that stuff available for free.

THE COURT: Well, why didn't Google use JDK back then?

MS. ANDERSON: Because, your Honor, by the time that Sun/Oracle had done this -- and there are a number other reasons we'll get into, I'm sure, at length.

Because, among other things, Google had already been investing in developing its own independent implementations of the Java APIs at issue or using other's independent implementations. It had already assembled an Android platform. It was getting ready to launch.

There was no reason to change course because, you know, at the time -- and we understand the Federal Circuit found differently, but certainly nobody considered the labels to be copyrightable, much less the use of them wouldn't be fair use.

I mean, these are -- these are, from the perspective of the industry, fairly outrageous to suggest that these labels that Sun/Oracle had been running around encouraging the industry to use, that Sun/Oracle reiterated over and over again: They can use those things. We just compete on the implementations. That's the words of their own CEO. They encouraged that. So that's fair use. And that is why once Google -- Google was already pretty much ready to release Android, and did later that year, there wasn't a need to change course.

And Oracle didn't change course until years later when it decided that it served its own purposes to try to take a 180-degree change of course in terms of their position on what

APIs are, whether they are protectable. That's a total change of course.

THE COURT: All right. I've got to let the other side have their say.

MS. ANDERSON: Thank you, your Honor.

THE COURT: Okay.

MS. HURST: Good morning, your Honor.

THE COURT: Good morning.

MS. HURST: Annette Hurst for Oracle on the technology tutorial.

Your Honor, I'll give short answers to several of the Court's questions and then I'll elaborate on them with a couple of slides, if that's acceptable.

In answer to the Court's question whether changing the names would still be an SSO violation, the answer is yes and the reason is that the SSO structure is independent of any specific name comprising that structure.

And I have a visualization of the SSO that I'm going to show the Court that's going to help understand that.

THE COURT: Okay.

MS. HURST: Another question the Court asked is whether anyone has used this analogously to Google without a license. And the answer to that question is no.

And, in fact, not just Oracle, but Sun enforced its rights in the APIs to preclude commercial use.

At one point the Government of South Korea was going to incorporate the Java API as part of a standard, a commercial standard for phones. And Sun actually got the U.S. trade representative involved, your Honor, to go to Korea and stop that from happening because it would result in wide-spread unlicensed commercial use of the Java API. And there are other examples of enforcement.

And, your Honor, we can also look to Google's expectations about the legitimate -- and I use that word as a quote -- uses of APIs under license.

Your Honor, before, just a month before Google acquired Android, it put up its own API for accessing its core Search service. And this is going to be Exhibit 5005, your Honor. It was not part of the last trial.

In that Terms of Use license for Google's own API to it's own Search service, Google said: Personal and legitimate uses only.

It said:

> "This is available to you for your personal non-commercial use only. You may not use Search results provided by the Google web API with an existing product or service that competes with products or services offered by Google."

That is what Google said about proper delineation between API usage for commercial and non-commercial purposes before it

acquired Android and came up with this whole argument.

And, your Honor, in that very document they assert a copyright interest in the API. They talk about the copyright notices, not removing them, and intellectual property rights, including intellectual property rights in the APIs.

So not just Sun enforced its right to preclude commercial use of its -- commercial unlicensed AP use of its APIs, but also Google.

Finally, your Honor, the other question the Court asked was: Why didn't Google use Open JDK back then?

Your Honor, the answer to that question is that it would not have been acceptable to the commercial carriers, the cell phone network wireless carriers and the original equipment manufacturers who had to adopt the Android platform. They both had to adopt the Android platform to make it work.

At that time Open JDK was under a form of the GPL, General Public License, and that GPL was widely understood to have what's called a viral nature. That means if it gets into your code and you use it in a certain way, it infects all the rest of your code and you have to publish it. That's the condition of that license. Once it's infected, you have to publish everything else that you did.

But the carriers and the OEMs didn't like that. Google didn't care. But the carriers and the OEMs didn't like that.

And the reason they didn't like it was because getting

down there in the guts of that code and getting infected with it meant they couldn't keep it secret and compete that way. The OEMs wanted to compete with those changes that they were making to the code on their hardware features.

And so document after document after document from Google will show during this period of time, your Honor, that they absolutely understood that this kind of a license was unacceptable to the commercial partners that they needed to launch the platform. And that is why they did not take Open JDK. And any efforts now to say otherwise are efforts to rewrite history, your Honor.

Now, in these packages -- going back to the specific technology piece, your Honor. In these packages there are two kinds of code: The declaring code and the implementing code. And both of them are in compiled form in the libraries. They are in compiled form in Android.

So not just the implementing code, but also the declaring code compiles and is present in the Android binaries that are installed in the nearly 4 billion phones, eReaders, cars, TVs and other devices. It's all in there. The declaring code, too, not just the implementing code.

Now, the API specification sets out just the declaring code piece, and that's just a part of the platform, and that's the work -- that's a piece of the work, overall work, that Google took. Google took that work in two forms.

MS. HURST: May I have the display system?

THE CLERK: Okay.

MS. HURST: Thank you, Madam Clerk.

(Document displayed)

MS. HURST: Google took the code in two forms.

First it took the literal elements, your Honor. And I think this goes to some of the questions that the Court was asking about earlier. It took literal elements that the Federal Circuit summarized as declaring code in the form of package names, class names, method headers, parameters and other elements.

And what I'm showing now, your Honor, is an example from a spreadsheet from one of our experts. The complete spreadsheet contains all the literal elements that Google took from a complete source code comparison out of the Java SE 5 API.

Your Honor, Google took more than 10,000 method declarations, which total comprised 11,429 lines of code. 11,429 lines of code.

And, your Honor, while Google is fond of showing the example of java.lang.math.max, the one that we've highlighted here is another example. It's a much more complex method declaration, your Honor. And what this method declaration does is invoke code to provide that security function, such as authenticating a certificate had been performed properly.

Now, that's a crucial function in a platform. Security,

we all know, is critical these days. And it doesn't matter if they didn't take the implementing code. That's not even part of the work. No man can excuse plagiarism by pointing to what he did not take.

The question is: What did they take? And this is where the Federal Circuit, your Honor, said they were overstepping on transformative. You can't point to all the stuff you added and say: I transformed it. You have to look at what they took and see whether they transformed that. And, your Honor, the evidence will show that Google did not.

Now, they took not just -- and, by the way, was this -- was 11,000 lines significant? This is a lot by any measure, your Honor. If the Court wants to look to other cases for benchmarks, an example is the *Brocade versus A10* case tried here in the Northern District a couple years ago in front of Judge Grewal. That case involved 145 lines of code out of 10 million and the jury returned a verdict of infringement and awarded half of the revenues, which Judge Grewal sustained on 50(b) and new trial challenges, your Honor. 11,000 lines of code is a lot by any measure.

Our experts will testify, your Honor, that other major accomplishments using fewer than 10,000 lines of code included the Apollo lunar module, the Canadian nuclear power plant emergency shutdown system. Think how important that has to be. The entire Unix operating system released in 1975 was less than

10,000 lines of code. It was 37 out of 166 packages. It's about 22 percent. 921 out of 3,257 public API classes. It was a lot and it can't be excused by pointing to what they didn't take. That's the law, your Honor.

Now we look at the SSO. Your Honor, I'm going to build up this SSO for you in elements.

(Document displayed.)

The first step in this slide shows all of the Java SE 5 public API this is every one of the classes in that public API and the interface. The blue dots are all the classes and the green dots are all the interfaces.

Your Honor, the 37 packages at issue are labeled with the black text. And so the Court can see in this visualization diagram where the 37 packages reside.

Now, your Honor, our expert, Dr. Schmidt, will testify that this type of a visualization diagram is reliably used to help with looking at structures in computer programs and other data structures in order to understand them and understand what has been taken. And so we start here at the highest level with the packages of the whole API and the classes.

Your Honor, the next step in structure shows the connections among the packages and interfaces. So you can see those lines creating the relationships, reflecting the created relationships that arise out of the declaring code, your Honor.

(Document displayed.)

This is now, your Honor, the complete set of relationships down to the class and interface level for the SSO. This doesn't include the method declarations. If we put that on there, we would have to slice it up in little pieces because the structure gets so complex, your Honor, that it's hard to discern.

But just sticking here, your Honor, you can see that the -- the blue dots, the green dots and the gray lines together comprise the Structure, Sequence and Organization. And this is the entire API, your Honor.

(Document displayed.)

Now, your Honor, this is the part that the Google took, the red lines overlaying the gray connectors, classes and interfaces. This is what it looks like if you remove the entire structure. Google took the beating heart of the Java Platform.

And, your Honor, the evidence from the experts will show that it was not just significant in a qualitative or subjective sense. There is now quantitative analysis called a centrality analysis, your Honor. This is a new technical analysis that Oracle will be offering.

The centrality analysis, your Honor, is based on an algorithm called Page rank. Your Honor, that's Larry Page rank of Google fame. It is the core of the significance algorithm used in the Google Search engine, and Google does not dispute

its use as a measure of significance. They can't.

Applying Page rank, your Honor, to the Java API classes, the 921 classes that were taken by Google, our expert will testify that the copied classes out of the platform were nine times more central to the work. Nine times more central to the work. And that's the test, the amount and substantiality of the portion taken from the work, not what was in Android. Nine times more important than what they didn't take.

And that was no fluke because the evidence will also show, your Honor, we did an analysis, this centrality Page rank analysis of the whole platform of Java, every single class. We took the top hundred most significant, most central ones and said -- Dr. Schmidt said: Dr. Kemerer, how many did Google take? Eighty of them. Eighty of the most important out of the top hundred.

And, your Honor, the evidence will show that it was important from a technical perspective to the OEMs and the carriers. They went on a roadshow to put together the Open Handset Alliance in 2006 and 2007. The very same time, by the way, your Honor, that SavaJE that Mr. Van Nest and Ms. Anderson mentioned, was failing. Google was out there promoting to the members of the Open Handset Alliance its new Android platform.

And the Court will see evidence that dozens of presentations that Google made during that process specifically touted the presence of Java APIs in the Android platform.

Specifically touted it. And the other internal emails show the same, your Honor; that they understood that was a way to leverage the existing base of developers and to get the carriers and OEMs to sign off on this.

So, your Honor, the centrality analysis, the visualization, Google's commercial conduct all point to the very great degree of significance of the 11,429 lines of code and the beating heart of the Java Platform.

Your Honor, our experts went beyond analyzing the significancy of the platform because we know that Google has tried to present the factors as if you should measure against Android. And so our experts did measure against Android.

The first thing that Dr. Schmidt determined is that if you remove any one of the packages, Android fails. None of the currently active 1.5 billion Android devices would work without Oracle's code in them. Period.

Dr. Kemerer looked at whether the applications are also dependent on the copied, unauthorized copied portions of the API. His findings, your Honor, are that the packages Google copied are very important for Android applications. Every single one of the top hundred Android applications depend upon at least three of the packages. The top 100 applications of all time for Android depend on an average of 11-and-a-half of the copied packages.

There are 14 applications, your Honor, that have been

downloaded in the Android store, the Google Play store more than a billion times, between 1- and 5 billion times. That's the biggest number of download categories that they have got there that they say in the store. There's 14 applications that meet that criterion. Those 14 applications rely on an average of 13.8 of the packages; a minimum of 8 and a maximum of 17. And, of course, there is Google's applications.

The technical expert evidence, your Honor will show that Google's most important applications -- Search, Gmail, Maps, YouTube -- all of them, your Honor, are dependent on these Java API packages; 15 in the case of Google Search alone.

The technical evidence will show, your Honor, that this literal element copying and Structure, Sequence and Organization copying was, therefore, very significant under Factor Three.

Your Honor, that leaves, I think, the issues of creativity and constraints.

Dr. Reinhold testified before, your Honor, that the API design is a very creative process and Google's witnesses agreed, both its fact and its expert witnesses.

Its former -- its Java guru testified API design is an art, not a science.

Former Google engineer referred to as Crazy Bob Lee said it's absolutely a creative activity.

Dr. Astrachan, Google's expert, likened it to being a

concert violinist. That's hard. It's hard in the same way.

Your Honor, in fact, the API specification in the declaring code are the most expressive part of the Java Platform because they are so valuable to developers because of their intuitive form, their ease of use. That is about expression, your Honor. That is the core of what the copyright is intended to protect. That's why Java was so popular in its early days.

Other object-oriented programming language, such as Smalltalk, didn't have APIs. When Sun proposed an API to go along with its language, and those were separately defined things, it became very popular because of its expression.

Now, Google --

THE COURT: All right. You're -- I think you are out of time.

MS. HURST: Thank you, your Honor.

THE COURT: You're over your time, but take one more minute to wind up.

MS. HURST: Your Honor, the API and the language are not the same thing. And the API is not a part of the grammar of the language. And it is misleading to say that the API must be freely available because the language was freely available.

THE COURT: I don't think -- I didn't hear Google make that argument. I don't think that's what they are saying. So I don't believe anyone is correcting that.

MS. HURST: Thank you, your Honor.

THE COURT: All right. We're going to take a short 10-minute break. When we come back, I want to give each side five minutes to raise with me one motion in limine that they propose to bring against the other side. Then I will give the other side five minutes to respond.

I'm not going to rule on it. It's not enough time, but I need to be -- it's going to help me start thinking about it so, you know, it won't come out of the blue later on. So it will be -- so five minutes, five minutes, five minutes, five minutes.

Then after that, time permitting, we're going to turn to the emergency motion that Oracle made concerning disclosure of documents to an expert.

So we'll take 10 minutes now. Thank you.

(Whereupon there was a recess in the proceedings from 9:43 a.m. until 10:01 a.m.)

THE COURT: All right. We'll start with Oracle. You can take five minutes to explain a motion in limine that you propose that you're going to be making in due course.

MS. SIMPSON: Good morning, your Honor.

THE COURT: Good morning.

MS. SIMPSON: The in limine that we would like to preview with your Honor involves Apache. And just to go back and give you some context for that. It did come up quite a bit

in the prior trial. Just to remind you of the facts that concern the Apache -- the Apache Harmony issue.

So Apache was a nonprofit company that wanted to do an independent implementation of Java and they set out to do that. But they did it under a license, your Honor. They got a specification license and they set out to create this code.

Once they had done that, they realized they had to come back and get what's known as a compatibility license, also known as a TCK license. And they came back to Sun and they asked for that license. And that resulted in a very large dispute between the companies.

There were public filings with respect to that dispute, and the industry all knew that there was a dispute over whether -- whether Apache could get the TCK license that it needed. The thrust that dispute was the fact that there were Field of Use risks in the TCK license. And the Field of Use restrictions in that license prevented someone who took that license from using the Java code in anything but a computer or a server. They couldn't use it on a mobile device. And so there was this public dispute.

Google was fully aware of this public dispute, your Honor. They participated in the public dispute. They even signed a letter that went to Sun about the public dispute about the TCK license and about the Field of Use restrictions. So they were fully aware that there was a license needed and that Apache

ended up not ever getting that license.

The fact that Apache was unlicensed continued to be a problem. They continued to have discourse with Sun. Eventually, as we've discussed earlier this morning, Sun open sourced Java and that essentially killed Harmony.

What ended up happening was IBM and Intel, companies that had been putting code into Apache Harmony, moved their code over to Open JDK and Harmony was then shelved or, as they called it, put in the attic.

At that time Apache also acknowledged that the Java specifications are proprietary. The quote that they put in in a public press release was the Java specifications are proprietary technology that must be licensed directly from the spec lead on whatever terms the spec lead chooses.

So this was a long and drawn-out dispute. The public and Google knew that there was a licensing issue here and that Apache was not free to ship Java without this TCK license.

THE COURT: But -- is it Simpson?

MS. SIMPSON: Yes.

THE COURT: Ms. Simpson, I'm drawing a blank on how -- how is it that Google wants to bring this in? What are they trying to make with this?

MS. SIMPSON: Correct, your Honor. That's where I was going right now.

There are three purposes that Google is going to seek to

use Harmony for. The first is -- or Apache. The first is this Apache Harmony story. They like to trot out the Apache Harmony story. Tell the story that this code was available out and Apache was using Java code. They did their own independent implementation and since everyone was fine with it, they like to say that, we were okay with it, they could do the same thing. So that's one thing they like to do with it.

The other thing that comes up, though, just to add to the confusion of this Apache -- Apache topic is that they also took the code from the Apache Harmony project. A lot of the code that they put into Android they did, in fact, take from -- or some of the code, I should say, they did, in fact, take from Harmony. So they can say that they have a license from Apache for the Harmony code.

Then thirdly, when they distribute Android, they distribute it under an Apache license.

So we've got three sort competing uses of licensing involving Harmony. And once you get -- once you look at all of that, you end up in a serious 403 issue here.

There is a very, very, very likely chance that the jury is going to be confused with respect to what was licensed, what wasn't licensed. They are likely to think that Google had a license for what it was doing.

And your Honor actually recognized this and was very, very concerned about the confusion last time. Apache was mentioned

every day of trial last time, the first trial. Every day the word Apache was mentioned.

Your Honor picked up on the fact that this is going to be very, very confusing, and you said at the charging instruction, quote:

"The danger that the jury will not understand how licensing works and that they will somehow think that there had been a license via Apache needs to be something that we're concerned about."

And you went on to say:

"You put a lot of evidence before the jury about Apache and all these people, and you're convincing me that I better put Apache in here by name. I think there is a risk the jury is going to think that you, Google, somehow had permission because you had a license from Harmony and you thought it was fine, and you didn't know that you needed a license, and all this stuff."

And then you say:

"I want to protect against the misimpression of off-track reasoning and the off-track reasoning would be that you had some kind of permission and it was okay to use it."

And that's the transcript at 2422.

THE COURT: That was very well said.

MS. SIMPSON: Yes, indeed, your Honor.

And the confusion bore out, your Honor. The jury sent five notes about Apache. Five.

THE COURT: Really?

MS. SIMPSON: Yes. And they all involve these exact questions. They asked: What are the terms of the Apache license? Did Google get an Apache license? If so, when? Were the 37 packages part of the Apache license? That's at ECF 1193.

These questions clearly show that the jury was highly confused; that this shed all kinds of difficulties into their deliberations and, really, here is the same problem.

THE COURT: You're almost out of five minutes.

But it does sound like some of the uses that you described would be legitimate uses. So what is your proposal for allowing Google to get the benefit of whatever there is there without just cutting it off all together?

Are you saying that whatever evidentiary value there is in the Apache Harmony, Google, just too bad for them, they don't get it at all? Or do they get some kind of proposed stipulation from you that -- I don't know. What --

MS. SIMPSON: Your Honor, I don't think there is much that can be done, right? This is a highly confusing situation.

There is also a timing issue. You heard me lay out the facts. Explaining those facts to the jury so that we could,

hopefully, dispel any kind of confusion and run through every single iteration of what happened with Apache will turn into a mini trial on Apache Harmony instead of what is supposed to be at issue here, which is whether what Google did is okay.

So the second part of your question involves relevance, your Honor. And I haven't touched on that yet, but there is a question of relevance here.

The law is very clear that it is not relevant to considerations what other third parties are doing. The Supreme Court said in *Petrella*:

> "It is not incumbent upon copyright owners to challenge each and every actionable infringement. The copyright owners can assess whether bringing suit is worth the candle."

So it's not relevant to this case whether we did or didn't go out after Apache. What we did with Apache with respect to Harmony is really not legally relevant.

THE COURT: Was the Supreme Court discussing fair use in that decision or were they discussing --

MS. SIMPSON: I believe they were discussing equitable estoppel.

THE COURT: Okay. Okay. Five minutes is up. Thank you. Good job.

Let's hear from the other side.

MR. KWUN: Your Honor, Michael Kwun for Google.

I just want to start with a little bit from *Wall Data versus Los Angeles County Sheriff's Department*. It's a Ninth Circuit case from 2006. It addresses fair use and then after listing what the four factors are but before going through an analysis of those four factors, the Ninth Circuit noted that:

"In considering and balancing the fair use factors, one must consider that fair use is appropriate where a reasonable copyright owner would have consented to the use."

And with respect to that they noted:

"i.e., where custom or public policy at the time would find the use as reasonable."

So with Apache Harmony there's a couple of points that I think tie into that.

First of all, Jonathan Schwartz testified in the prior trial that there was no reason that Apache could not ship Harmony. And Harmony was -- Apache Harmony did not have a specification license. They did not have any license at all. And I believe that --

THE COURT: I'm sorry. Who testified to that?

MR. KWUN: Jonathan Schwartz, former CEO of Sun.

THE COURT: He's the one that did the blog.

MR. KWUN: Yes.

THE COURT: Okay.

MR. KWUN: And he was CEO at the time of the dispute

that Ms. Simpson referenced.

So if the question is whether what a reasonable copyright owner would have done and whether they would have consented to the use, the fact that --

THE COURT: When he said that, did he know all the facts? In other words, was he misinformed in some way about what the situation was or did he -- was he fully informed and nevertheless said they didn't need a license?

MR. KWUN: The Harmony code was developed in the open. It was open source. It was -- there was a continuing dispute over branding, whether or not Apache could get the TCK license and that would allow them, if they passed it, to say that our Apache Harmony code is Java code. It's Java brand code. They could have used the coffee cup logo and this would allow people to realize that this code is going to do exactly what Sun says it's going to do.

So over the branding, that was a separate issue. He said they can't use the branding unless they get a TCK license and if they get the TCK license, it will have field of use restrictions. But without the TCK license they can ship today, they just can't call it Java.

So that's what he said. He was fully aware of exactly what the situation was.

THE COURT: Okay.

MR. KWUN: So that's one point.

The other point is that there were a lot of contributors to this --

THE COURT: But tell me how you plan to use the Apache Harmony? What are your arguments to the jury going to be based on that?

MR. KWUN: Well, so that ties into the next point, which is there were contributors to the Apache Harmony code that were major industry players. Intel contributed code. IBM contributed code. They did this because they were using Apache Harmony in their products. Not all their products, but in their -- there were releasing products that used Apache Harmony code.

What this shows is that the custom or public policy at the time defined the use as reasonable. Both third parties, Sun's own CEO, everyone viewed this use -- which included, of course, far more than the 37 API packages, included all of the API packages from Java SE, that the industry, that custom and public policy at the time defined the use as reasonable. And under Ninth Circuit law, that is a factor to consider in balancing the factors.

So separately there has also been mention of the Apache license that Android is offered under. We need to be able to provide context to the jury to explain how we distribute Android, what our business model is. And part of that is that we use a license that comes from the Apache Foundation. It is

a license that exists independent of the Apache Harmony code.

So the Apache Foundation, they created a software license that anyone can use for their software. Google uses it for a variety of software that has nothing to do with Apache Harmony, has nothing to do with Java. It also uses it for most of the elements in the Android platform. And it uses it for the core libraries, which are at issue in this case, but it also uses it for other aspects of Android that were developed entirely separately from the Apache Harmony project.

So if we are not allowed to have to talk about the Apache license, we are going to be unable to describe elements for a business practice.

So I do think that there are -- A, I think that we need to be able to talk about Apache Harmony.

But, B, I think that the separate issue of the Apache license, they can't be put together simply because they both have the word "Apache" in them. They both come from the same non-profit organization, but they are actually two separate concepts.

Unless your Honor has any other questions.

THE COURT: But if -- this is an if. But if Apache needed a license from Java and did not get it, then the fact that Google got a license from Apache would not be a defense against a direct suit by Oracle for using Java even if it's exactly the way it's used in Apache; true?

MR. KWUN: We agree. We are not trying to say that the jury can short circuit the entire deliberation process by saying: You have a license from the Apache Foundation to Apache Harmony and, therefore, it doesn't matter whether there is a fair use or not.

But the -- whether or not Apache had -- the fact that Apache did not have, itself, a license from Sun demonstrates that significant players in the industry, including all the people who contributed code to Apache Harmony, were acting under the impression, under the custom and public policy at the time, that what they were doing was okay.

THE COURT: Did Google know that at the time?

MR. KWUN: Who the contributors were? Yes.

THE COURT: But did Google know there was no license?

MR. KWUN: Yes.

THE COURT: Are there documents that back that up?

MR. KWUN: There is going to be at least testimony and possibly documents.

THE COURT: All right. Okay. Thank you.

Ms. Simpson, I'll give you one minute to respond to this and we'll move to something else.

MS. SIMPSON: Your Honor, I would just start with Mr. Schwartz's testimony. He also testified that he did not know what the legal requirements were of the license and that he hadn't looked at them, and so his testimony was entirely

unrelated to a legal analysis of what the licenses required.

With respect to Google's suggestion that they need to refer to these things because of the custom, industry and practice, the law is very clear and the Federal Circuit said that you -- just because something is in practice or common or popular, doesn't mean that it loses protection for copyright and doesn't mean that you just get to take it. It doesn't become fair use because it's become popular.

THE COURT: Okay. Thank you.

All right. So now Google gets to bring up one motion in limine and preview it.

MR. PAIGE: Good morning, your Honor. Gene Paige for Google.

And the motion in limine we would like to discuss today in preview is a motion in limine excluding testimony and evidence relating to implementations of Android, such as Android TV, Android Auto, Android Wear, Brillo or other such implementations.

Now, your Honor issued an order on February 5, 2016 instructing that the trial will not include implementations of the Android and the Android TV, Android Auto, Android Wear or Brillo.

Nonetheless, after that order, we received an expert report from original by Dr. Adam Jaffey that spent page after page discussing various market failures by Oracle in the areas

of wearable devices, automotive devices, the internet of things, VOIP phones, household appliances, printers, Smart TVs. So those went from Paragraphs 366 through 371, 386 through 391, 392 through 395, 396 to 405. Just paragraph after paragraph discussing these items.

And so Google is concerned that Oracle may intend to still present evidence on such devices despite the Court's order.

Now, given the limited scope of the retrial and the time available for the presentation of evidence, it would be an inefficient use of the Court's and jury's time to put on evidence about these other things where there is a need to show that, in fact, they are using the 37 APIs that are at issue. And it would be prejudicial to Google to allow that kind of testimony without evidence that they are using the 37 APIs because then the jury might well assume that these things are used in the 37 APIs without any proof being given, and the jury could mistakenly believe the reason that Oracle is allowed to discuss them is because we concede that they are being used in 37 APIs in these things.

And just as an example, Brillo is one thing that they have talked about. And in their expert report from the Mr. Kemerer, their opening expert report, what they said about the use of the 37 APIs in Brillo was that he looked at the source code computer, where Brillo's source code was found, and he found certain directories on that source code computer and those

directories had things that appeared to be the class files used in the 37 APIs.

But there is no testimony that -- or evidence from Dr. Kemerer that, in fact, these things are used. His evidence of use by Brillo appears to be just that these files are found on the same source code computer.

We don't think that's enough to show use by Brillo and our experts would disagree with that. But since, your Honor has said that trial won't include implementations of Android on Android TV, Android Auto, Android Wear or Brillo, we don't think it's proper to have any evidence of that come in at all.

THE COURT: Okay, thank you.

Let's hear from the other side.

MS. SIMPSON: Your Honor, this evidence is critical to our Factor Four fair use presumption. And although your Honor did issue an order regarding infringement, that order was with respect to what items were infringed and what items could be infringing. And that order came out of a motion to strike that was directed expressly at Java 6 and Java 7. It didn't talk about --

THE COURT: I want to be very clear. I said we were going to go back to where we were in the first trial and replay that trial as if I had the benefit of the Court of Appeals. The only finding of infringement in this case was with respect to a handful of things like -- I think they are called Chrome.

I can't remember all those names any more, but those were found to be infringed. TV was not found to be infringed.

MS. SIMPSON: Understood, your Honor.

THE COURT: So we're not going to have an infringement trial here on -- that's for another trial later on in the future.

MS. SIMPSON: Right.

THE COURT: So how can you possibly get into that stuff without trying -- you'd have to prove that it infringed. And I don't want -- we're not going to do that.

MS. SIMPSON: Understood, your Honor. We are not offering it for infringement in light your order.

THE COURT: But it's not relevant unless it does infringe.

MS. SIMPSON: Your Honor, it is entirely relevant, and let me explain to you why. Let me explain to you why it's relevant.

THE COURT: All right.

MS. SIMPSON: The fourth factor of market harm is a forward looking test. It requires you to look both at current markets and, also, potential markets, markets that you might be in, and also what happens if the use becomes widespread. That's a hypothetical test into the future.

All right? This information, these critical deals that were lost and these markets that we were in, they are all

relevant to fair use without infringement. We don't need to -- you know, we need these -- even if these were never part of the infringement discussion, these markets would still be relevant to fair use.

And I can read from the Federal Circuit decision that says it requires -- and they are quoting *Campbell* from the Supreme Court:

> "Fair use focuses on the effect of the use on the potential market for the copyrighted work. It requires Courts to consider not only the extent of the market harm caused by the particular actions of the infringer, but also whether unrestricted and widespread conduct of the sort engaged in by the defendant would result in a substantially adverse impact on the potential market for the original."

So these markets that we're talking about -- and the Jaffey puts these in on the fourth factor of harm. These markets are examples of places where Java was and Android is now taking our market share. Some of those involve TV and some of them involve auto, but some of them involve tablets and eReaders. There is a whole list of products. They are not necessarily related to the four that were listed in the order.

There is IOT. There is Voice Over IP phones. There's home entertainment. GPS systems. Vending machines. Printers.

THE COURT: Aren't you going to at least have proof

that those areas, those products use the copyrighted -- I don't see how you get around -- why is it even relevant unless it uses Java?

MS. SIMPSON: Your Honor, the use that's being focused on here is the use in Android. And your Honor has already decided that the certain versions of Android, all the way up through Marshmallow. Lollipop and Marshmallow, those are the things that we're looking at to see whether they are causing harm.

It is the use of Lollipop in a phone that's causing us harm. It is the use of Lollipop in a TV that's causing us harm.

THE COURT: But you have to prove that Lollipop an used in the TV.

MS. SIMPSON: Your Honor, I don't think they dispute that Lollipop is used in these other products.

THE COURT: You've got to prove it. It's your burden of proof to prove that part of infringement.

MS. SIMPSON: Well --

THE COURT: And I've said and I say we're not going to have an infringement trial.

MS. SIMPSON: Your Honor, I -- I understand --

THE COURT: I'm having a fair use trial and damages and willfulness. We've got enough on our plate. I don't know.

MS. SIMPSON: Your Honor --

THE COURT: Let me hear from the other side a minute on something.

MS. SIMPSON: May I just make one more point on fair use, your Honor?

This is not only relevant to the fourth factor. It's also very relevant to the first factor and on the transformative use.

We've talked a little bit about substitutions and the first factor requires you to look at whether this -- the use that Android is make is a superseding use.

These markets, these places where Android has taken place of Java in certain products is the exact example of a superseding use. When you have a superseding use --

THE COURT: But Java never went there in the first place. What do you mean, it's taking place?

MS. SIMPSON: No, we --

THE COURT: Java never went to TV.

MS. SIMPSON: We have evidence that we were. We have evidence that we were in the loss of these markets and that's what we -- that's what the Jaffey report does, our expert.

THE COURT: I can't believe that. You're telling me that Java was in the TV market.

MS. SIMPSON: It was indeed, your Honor.

THE COURT: Let's hear from -- Mr. Paige, what I'm hearing is from Ms. Simpson that they don't need to prove

infringement at all in order to get into the TV thing because it's just independently relevant to the four factors.

Now, is that right or wrong? What is the answer to that?

MR. PAIGE: I don't think that's right, your Honor.

The Brillo example that I just gave is something where I think there is absolutely no question that what they put forward can't prove infringement. They have just said these things --

THE COURT: But their argument is, we don't even have to prove it would be infringing. It's just -- if they hadn't infringed on these lesser products, then we would have grabbed this gigantic market out there involving TVs and it doesn't matter whether the TV infringes or not. We got cut out of that market due to the intermediate infringement by Chrome.

MR. PAIGE: I think that does require proof, your Honor, especially if what they're talking about --

THE COURT: In your motion you better prove that part up; that in a sina qua non is -- as a proof of that fair use argument, is that they prove infringement.

MR. PAIGE: Especially if what they are talking about in the use of TV is Java ME, which doesn't have all of the 37 APIs, right?

I don't know what they are claiming the TV is using when Java was supposedly in TVs. But if it's not using SE, the copyrighted work that it has the 37 APIs, I'm not sure how it

would be relevant.

THE COURT: Well, what Ms. Simpson said is the jury has already found that Google infringed subject only to fair use on these -- give me the names again of the different operating systems?

MR. PAIGE: Honeycomb, Ice Cream Sandwich, Kit-Kat, Lollipop, Jelly Bean.

THE COURT: Right. Those are called operating systems? What are they?

MR. PAIGE: Called versions of Android, sure.

THE COURT: All right. Versions of Android.

That if those versions of Android had not used the 37 APIs and the SSO, then somehow Sun and Oracle would have leapfrogged into the TV market and now enjoy that dominant position that you are enjoying. And it doesn't even matter whether or not the TV uses Java or infringes in any way. It's just sort of a but-for thing. They would have grabbed that?

I think that's your argument, right?

MS. SIMPSON: Well, your Honor, it's --

THE COURT: Otherwise, you've got to prove infringement and I don't think we're going to do that. We're not going to get into infringement.

MS. SIMPSON: Your Honor, we're in an unusual place here because this case has been going on for some time. And typically when you're doing this fair use analysis, you're

really stuck with looking at potential markets, markets that haven't yet been formed. And the law is very clear that you're allowed to look at markets that might happen. You're allowed to look at markets that people don't even necessarily think they are going to go into.

So I'm not saying --

THE COURT: Are those markets where the copyrighted material is the -- is what's used?

I mean, how can you possibly benefit -- I mean, we're talking about copyrighted material here. You're trying to say that you would get profits that have nothing to do with the copyrighted material.

MS. SIMPSON: Your Honor, this has nothing to do with profits. I just want to be clear. We did not include these products in our disgorgement analysis. I'm talking entirely about fair use. All right?

And we're allowed to look at future hypothetical markets. And, yes, it has to be reasonable that Java would be in those markets. And we intend to present evidence that will show that it would be reasonable that we would be in these various market segments. And that's what we're required to do.

And if the Android is now --

THE COURT: But then you're going to have to prove that these other markets, in fact, used the copyrighted material at issue.

MS. SIMPSON: Your Honor, we would if we were proving actual harm, yes. But we're also allowed to prove future harm and potential harm.

THE COURT: I don't know. I don't know how -- at some point Rule 403 has got to come in here and keep this trial from being a zoo.

I don't know if we're going to go down that path. I am not going to say no yet. I've got to think about it and see your real motion, as opposed to the preview.

But this disturbs me that we would be getting into these extended -- all right. You both did a great job. Thank you.

MS. SIMPSON: Thank you.

MR. PAIGE: Thank you, your Honor.

THE COURT: Let's focus for a moment on this more recent -- I have the -- all right. Do you want to -- have you two resolved this issue?

MR. BAYLEY: We have, your Honor.

THE COURT: You have?

MS. SIMPSON: Yes, your Honor.

THE COURT: Come tell me what the resolution is.

THE REPORTER: Counsel, your name, please.

MR. BAYLEY: Edward Bayley.

THE COURT: All right. Tell me the resolution.

MR. BAYLEY: So the resolution, your Honor, is based on some additional information and representations we received

outside the courtroom, Google is willing to withdraw its objection as to Dr. Toubia.

THE COURT: That's great. Thank you.

I have a couple of suggestions I have for you. And these are not rulings, so don't quote these back yet. But we have a big case to manage and I just want you to know, as a general proposition, I think you should both be thinking that anything more than 24 months old is stale and should not be under any kind of *Attorneys' Eyes Only or* somehow protected.

Now, there can be some minor exceptions to that. So I recognize -- but as a blanket rule, with minor exceptions, I think you ought to be thinking that anything that is that old in the industry you're in is stale and does not deserve some sort of special treatment. So that's number one. I'm not ruling that. I'm committing it to you for your consideration.

So it would work both ways. Both of you would have this -- this rule. Not rule, but suggestion.

The other is this. I'm disturbed by the idea that some expert can get on the stand and conceal from the jury their sources of bias and such. Oracle puts on the stand somebody who may have a very huge bias because of their prior commitments and their prior work and then refuse to tell the jury that because they are under some confidentiality agreement. I don't like that. I'm sure that you have the same problem.

MR. BAYLEY: Yes, your Honor.

THE COURT: And I don't like that either. I'm just telling you, it's not plausible. I think the way you big firms and big companies think that you're going to be able to come in here and lead the jury like a ring in their nose through hard gun experts, it is so disturbs me that that should not be the way cases are tried.

The jury is capable of making a lot of these decisions without the benefit of high paid experts, especially if they are going to conceal facts from the jury.

I'm not making any ruling. I'm just telling you that since I was a trial lawyer, you law firms have changed and the use of experts has gotten out of hand, in my view.

I hope that you each have about two or three experts, that's it. But I'm getting the sense that you have seven, eight, ten, twelve, experts.

How many experts do you have?

MS. SIMPSON: Six.

THE COURT: How many?

MS. SIMPSON: Six? Seven.

MR. KWUN: I believe it's seven with Dr. Toubia.

MS. SIMPSON: Now we have seven.

THE COURT: Seven, all right.

And how many do you have?

MR. BAYLEY: Sorry, your Honor. I can't remember off

the top of my head. I believe it's four or five.

THE COURT: See, there are so many you can't even count them.

How many?

MR. BAYLEY: Five. I'm being told five.

THE COURT: And that includes all phases of the case?

MR. BAYLEY: Yes.

THE COURT: Five is a little higher than I would like, but okay. Seven I -- that's really pressing the outer limits. I'm not saying no to that yet, but at some point we've got to stop putting -- laying so many experts on the jury.

Okay. But I thank you for resolving this, this problem. We'll go forth, do good. I have benefited very much by this presentation and I just want to ask one last question.

When is the next time I will see you in this case?

MR. VAN NEST: I think for the pretrial, your Honor. It's in April.

MS. SIMPSON: April 27th, your Honor.

THE COURT: Are there any -- okay. Pretrial in April. All right. Okay. So maybe a number of weeks great. I think we're done.

MS. SIMPSON: Thank you, your Honor.

MS. ANDERSON: Thank you, your Honor.

(Proceedings adjourned.)

CERTIFICATE OF OFFICIAL REPORTER

I certify that the foregoing is a correct transcript from the record of proceedings in the above-entitled matter.

Debra L. Pas, CSR 11916, CRR, RMR, RPR

Friday, February 26, 2016